Ranking the Uniformity of Interval Pairs

Jussi Kujala and Tapio Elomaa

Department of Software Systems
Tampere University of Technology
P. O. Box 553, FI-33101 Tampere, Finland
jussi.kujala@tut.fi, elomaa@cs.tut.fi

Abstract. We study the problem of finding the most uniform partition of the class label distribution on an interval. This problem occurs, e.g., in supervised discretization of continuous features, where evaluation heuristics need to find the location of the best place to split the current feature. The weighted average of empirical entropies of the interval label distributions is often used in this task. We observe that this rule is suboptimal, because it prefers short intervals too much. Therefore, we proceed to study alternative approaches. A solution that is based on compression turns out to be the best in our empirical experiments. We also study how these alternative methods affect the performance of classification algorithms.

1 Introduction

We consider the problem of processing labeled and sequential data into intervals — contiguous subsequences — that can be utilized in prediction. This task is encountered, e.g., in the discretization of numerical attributes when learning classifiers. Top-down greedy heuristics reduce this problem to a simpler one [1]: How to rank and compare the uniformity of two adjacent intervals?

Arguably the most often used measure of uniformity of adjacent example intervals is the weighted average entropy over the class label distributions [2]. Using entropy is a well-founded approach that should in principle lead to good results. We cannot, however, compute the true entropy underlying the label distribution. Instead, we need samples to estimate it through the observed empirical entropy.

We will evaluate the suitability of the empirical entropy in finding the least uniform intervals. Our empirical evaluation shows that the estimation of entropy often fails in this task, because it prefers too short intervals.

A similar and related problem occurs in the top-down induction of decision trees, where the tree-building algorithm searches for the most informative attribute. It is known that criteria based on empirical entropy prefer too much attributes with several values. On the other hand, in the interval selection problem we only have two values for the attribute — the left and the right interval. Hence, the failure does not occur because of too many values for the attribute. See the previous work discussed in [3] for more information on attribute selection in decision trees.

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part I, LNAI 5211, pp. 640–655, 2008.

The reasons behind this suboptimal behavior of empirical entropy in decision trees and in splitting an interval are similar. First, the estimation error of entropy is significant, and even more so when fewer samples are available. In particular, this error is biased towards intervals with less data, and some interval pairs *always* contain a short interval, because we consider *all* splits of a larger interval. Second, perhaps a more minor reason is that minimizing entropy does not necessarily coincide with minimum empirical error. Kohavi and Sahami [4] discuss entropy-based and error-based discretization more generally.

In this paper we evaluate three simple approaches intended to rectify the problems of the approach based on empirical entropy. The methods also take into account the number of samples that is available to estimate the distribution. Previously, only the empirical frequencies have been used. We evaluate these three approaches by generating synthetic data from known distributions and observing the distributions of the resulting split points. Our empirical evaluation demonstrates that in terms of reducing the absolute error, the proposed approaches are successful, but their utility in improving prediction error of Naïve Bayes (NB) is smaller.

In the next section we introduce the required preliminaries for the rest of the text; the NB classifier and the recursive entropy heuristic. Section 3 illustrates the failure of average empirical entropy in always choosing the best split point. We also provide a theoretical explanation for this shortcoming. In Section 4 three approaches that try to overcome the problems are put forward: The first approach is based on choosing the split point that yields the best compression of class labels. We can also consider it as maximizing a certain posterior likelihood. In the second approach the maximum likelihood estimation of probability in entropy calculation is replaced by a Bayesian estimate. The third approach replaces entropy with another function with different concavity properties. Section 5 evaluates the proposed approaches empirically. The implications for classification of the approaches studied in this paper are the topic of Section 6. In particular, we consider NB and test the implications also empirically. Finally, we put forward the concluding remarks of this work.

2 Preliminaries

Let us first introduce the NB classifier and then discuss how it relates to discretization and finding non-uniform intervals. Let y denote a variable that takes a value of a class label, and let x denote a vector of d features $\langle x^1, \ldots, x^d \rangle$. We are given a set of n examples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. In the NB classifier we assume that the features are statistically independent given the class, which results in the following probability for a label y given a vector x:

$$\mathbf{P}(y \mid x) \propto \mathbf{P}(y) \prod_{i=1}^{d} \mathbf{P}\left(x^i \mid y\right),$$

where x^i are the independent features. The NB classifier then selects the label with maximal probability. Note that although the statistical independence is

Table 1. The EMP-ENT split point selection method and the recursive entropy heuristic

---

Function EMP-ENT
Input: An interval I.
Output: Two subintervals (I_1, I_2) which partition I.
Algorithm: For all class labels y_j, $j = 1, \ldots, m$, let $\widehat{\mathbf{P}}_I(y_j)$ stand for the *empirical probability* of observing label y_j on interval I. In other words, it is the ratio of labels y_j to all labels in interval I. Now, the *empirical entropy* of the class label distribution of I is

$$\widehat{H}(I) = -\sum_{j=1}^{m} \widehat{\mathbf{P}}_I(y_j) \lg \widehat{\mathbf{P}}_I(y_j).$$

Let $|I|$ denote the number of examples in interval I. The average empirical entropy of a particular split (I'_1, I'_2) is

$$\frac{|I'_1|}{|I|}\widehat{H}(I'_1) + \frac{|I'_2|}{|I|}\widehat{H}(I'_2).$$

Return the split that minimizes the average empirical entropy.

Function RECURSIVE ENTROPY HEURISTIC
Input: An interval I.
Output: A contiguous sequence of subintervals $(I_1, \ldots, I_k)$ which partitions I.
Algorithm: Out of all splits (I'_1, I'_2) select the one given by the EMP-ENT method. If a stopping criterion, such as the MDL rule used by Fayyad and Irani [2] is satisfied, then return only I. Otherwise, return a concatenation of outputs of recursive calls to the recursive entropy heuristic with inputs I'_1 and I'_2.

---

used to derive the decision rule, the NB classifier does not necessarily fail if features are correlated, because it works as long as the correct label has maximal probability [5,6,7].

For discrete features the marginal probabilities $\mathbf{P}(x^i \mid y)$ are easy to estimate by counting from the training examples. One usually smooths the count e.g., with a Laplacian estimate, to prevent assigning the zero probability to some events. Continuous features are more difficult to deal with, but there are several solutions for handling them, none of which appear to dominate the other [8].

Discretization is a solution in which an interval is divided into discrete bins, and the marginal probability is estimated by counting the items that fall into these bins. This approach has attracted significant attention [9]. The best performing methods are founded on recursive entropy heuristic [1,2]. Its aim is to minimize the empirical entropy of the bins, while avoiding creating adjacent bins that appear to come from the same underlying distribution. For k bins, the number of possible bin borders for n items is $\binom{n-1}{k-1}$, which scales in $\mathcal{O}(n^{k-1})$ (we do not accept empty bins in this count). Hence, a brute force approach to the combinatorial explosion is unattainable.

Thus, the recursive entropy heuristic uses the greedy top-down approach. In it one successively splits in two the interval yielding minimum entropy until some stopping criterion is satisfied. This heuristic is detailed in Table 1.

3 How Minimizing Empirical Entropy Can Fail

We have observed empirically that the EMP-ENT rule often proposes using a very short interval — from one to five items. This behavior appears to be a result of random noise in the labels, rather than a correct decision. In this section we first discuss the suitability of using the entropy in finding a location to split, and then proceed to study how empirical estimation of entropy is difficult.

3.1 The Role of Entropy in Selecting Uniform Intervals

The 0/1-loss is the probability of predicting a wrong label. It gives a simple method to select the split location: minimizing the 0/1-loss of the resulting subintervals. In this case we predict the majority label on each subinterval. However, 0/1-loss is blind to those differences in the label distribution that do not change the majority label [9,10].

For example, if the most likely label remains the same on the whole interval, then the 0/1-loss of all possible splits is the same constant. Naturally, if we use these subintervals to predict labels under 0/1-loss, then it is not useful to split the interval in this case. Nevertheless, if we combine the prediction from this and another feature, then we would like the combined predictor to perform well. In this case the interval should be split at the location where it provides the least error for the combined predictor.

This is the motivation for minimizing the joint entropy of subintervals, it can distinguish changes of the label distribution. For example, let $p(x)$ denote the probability of generating the label a at the position x on the interval, and let $1 - p(x)$ be the corresponding probability for the label b. Then the analysis of the Lagrangian reveals that the entropy is minimized only in those locations in which either the majority label changes or the derivative $p'(x)$ is zero.

However, minimizing entropy does not necessarily minimize 0/1-loss, although these two often coincide in practice. Figure 1 illustrates a situation in which the entropy prefers to split at a location where the resulting subintervals are non-uniform. The 0/1-loss, however, is minimized at another location. Hence, using entropy is not justified if we, for example, have a single feature and want to select a single split point on it for a further use in the NB classifier.

Note also that, although the first split point may minimize 0/1-loss, the second does not necessarily give the minimum 0/1-loss for two split points. In the situation of Figure 2 the entropy is minimized in the middle. However, the two split points that minimize 0/1-loss are at the locations in which the most likely label changes.

Therefore, both 0/1-loss and entropy fail in some sense. The loss is blind to some differences in the label distribution, and the entropy fails to provide the

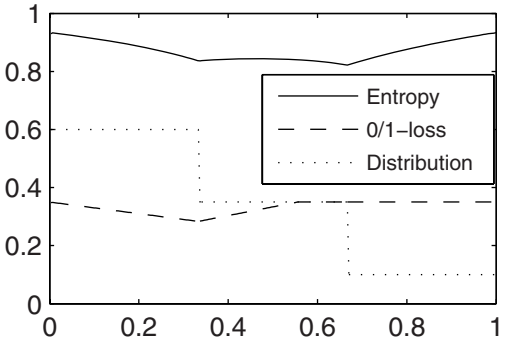

Fig. 1. Entropy and 0/1-loss as functions of a split point on the interval $[0, 1]$. The distribution that generates labels a and b is also shown. The probability of label a changes from 0.6 to 0.35 at the location 0.33 and to 0.1 at the location 0.66. The optimal split point is at the location 0.66 according to entropy, but at the location 0.33 according to 0/1-loss.

minimum 0/1-loss. However, in experiments the entropy usually performs better than minimizing 0/1-loss [4]. Two facts could provide an explanation. First, the problem domains have several features, so several features jointly interact in prediction. Second, due to the recursive nature of the heuristic, the splitting continues until 0/1-loss is minimized or nearly minimized. A small data size might be a problem in this case.

3.2 Empirical Estimation of Entropy

Although entropy is an acceptable measure, we cannot use it directly, because it depends on the hidden underlying distribution. Instead, we have a sample from this distribution. It is known that entropy is difficult to estimate; for instance, there is no unbiased estimate for it [11]. More information on the estimation of the entropy is given, for example, in [11,12].

Let us first give a simple demonstration of using the empirical entropy in our application of splitting an interval and, then, a reasoning behind the observation. Figure 3 illustrates the distribution of the optimal split point according to EMPENT rule of Table 1 on two separate class label distributions. In both cases thirty class labels were drawn so that class label a initially has probability $1/3$ and after the change point probability $2/3$. The change point in distribution A is located in the middle and in distribution B after the fifth item. Ten thousand random intervals were drawn from both distributions.

For both label distributions the correct change point is clearly the most often identified split point location, but there are also noteworthy concentrations at both ends of the interval. These concentrations have no particular justification, they are just bogus local optima. In Section 5 we observe that this behavior occurs, regardless of the interval length, if the optimal split point location of the distribution is unclear.

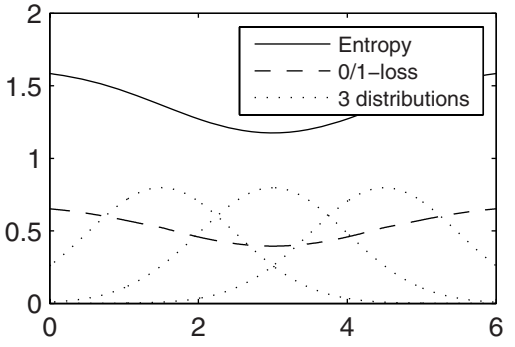

Fig. 2. Three Gaussians generate three labels. Both entropy and 0/1-loss are minimized in the middle. For two split points the 0/1-loss is minimized at the locations, where the most likely label changes.

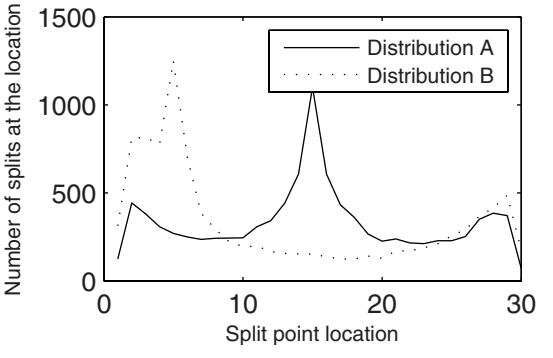

Fig. 3. The optimal split point distribution according to the EMP-ENT rule, for two different distributions of the intervals. Ten thousand random intervals were generated.

Let us consider the reasons for the phenomenon illustrated in Figure 3. Let $\boldsymbol{p}$ denote the real distribution on an interval under consideration and let $\hat{\boldsymbol{p}}$ denote the empirical distribution of a draw from $\boldsymbol{p}$. The empirical entropy $\widehat{H} = H(\hat{\boldsymbol{p}})$ is a random variable of labels drawn from $\boldsymbol{p}$. Two factors contribute to the difference between $H = H(\boldsymbol{p})$ and $\widehat{H}$ [13,6,14]:

1. The *bias*, which we define as

$$H\left(\boldsymbol{p}\right) - \mathbf{E}(H\left(\hat{\boldsymbol{p}}\right)).$$

2. The *variance*, which tells how much $H(\hat{\boldsymbol{p}})$ changes around its expectation:

$$\mathbf{E}\big((H\left(\hat{\boldsymbol{p}}\right) - \mathbf{E}(H\left(\hat{\boldsymbol{p}}\right)))^2\big).$$

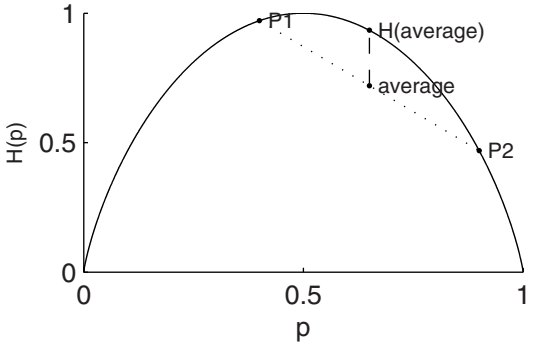

Fig. 4. The concave binary entropy function. Note how the expectation (average) of entropies at points $P1$ and $P2$ is less than the entropy of their expectation.

It is well known that the *mean squared error* (MSE) of estimation is related to these values [6]:

$$\underbrace{\mathbf{E}\left(\left(\widehat{H}-H\right)^2\right)}_{\text{MSE}} = \underbrace{\left(\mathbf{E}\left(\widehat{H}\right)-H\right)^2}_{\text{bias}} + \underbrace{\mathbf{E}\left(\left(\widehat{H}-\mathbf{E}\left(\widehat{H}\right)\right)^2\right)}_{\text{variance}}.$$

Both the bias and variance are affected by the concavity of entropy $H(\boldsymbol{p})$. Jensen's inequality [15] asserts that for any concave function f it holds that

$$\mathbf{E}(f(X)) \leq f\left(\mathbf{E}(X)\right).$$

An immediate consequence is that

$$\mathbf{E}\left(\widehat{H}\right) = \mathbf{E}(H\left(\hat{\boldsymbol{p}}\right)) \leq H\left(\mathbf{E}(\hat{\boldsymbol{p}})\right) = H\left(\boldsymbol{p}\right) = H.$$

For example, the empirical entropy of one sample is always zero.

Another complication is the variance of the empirical entropy. Let $\hat{\boldsymbol{p}}_{\text{ML}}$ be the maximum likelihood estimate of $\boldsymbol{p}$. The estimate is unbiased, but it naturally has a high variance with small sample sizes. This variance can result in larger than expected deviations in $\widehat{H}$ for small values, because the entropy H grows fast around a non-uniform distribution. For example, if we draw 30 labels from a distribution over the labels $\{a, b\}$ with $\mathbf{P}(a)$ equal to $1/5$, then $H = 0.72$ and $\mathbf{E}(\widehat{H}) = 0.7$. Although these values are close to each other, the probability that $\widehat{H} > \mathbf{E}(\widehat{H})$ is 0.57. This implies that low values for $\widehat{H}$ are further from $\mathbf{E}(\widehat{H})$ than high values for it.

Of course, both the bias and the variance tend to zero as the number of samples grows, because the maximum likelihood estimate $\hat{\boldsymbol{p}}_{\text{ML}}$ concentrates around a point and the entropy is almost linear in the neighborhood of this point. Jensen's inequality is strict when the concave function f is linear and, in fact, the difference $f(\mathbf{E}(X) - \mathbf{E}(f(X)))$ depends on the "curvature" of f. Figure 4 demonstrates this graphically.

4 Alternatives to Empirical Entropy

We now study three alternative solutions to the problem(s) identified in the previous section. They differ in their justification and operation. We will study their performance empirically in the next section.

4.1 COMPRESS: Compression

There is another potential justification for the empirical entropy, compression. The MDL principle, roughly, advocates the selection of the model that compresses the data the most [16].

If we know $\hat{\boldsymbol{p}}$, then $H(\hat{\boldsymbol{p}})$ is approximately the expected number of bits needed to encode labels from the distribution $\hat{\boldsymbol{p}}$. However, this is not equal to compressing the interval I, because we know the *exact* number of labels, not just their probabilities. In addition, $\hat{\boldsymbol{p}}$ is unlikely to be close to the true distribution for small sample sizes.

If we know the empirical frequencies of the labels on an interval I, then we can compress it by identifying the permutation that transforms a known interval to I. The number of permutations of the labels on I is

$$\mathbf{Perm}\,(I) = \binom{n}{n_1, \ldots, n_m} = \frac{n!}{n_1! \cdots n_m!},$$

where n is the number of items on the interval I and n_i is the number of items with label i. Therefore, the interval I can be identified with $\lg \mathbf{Perm}\,(I)$ bits, if we know the original permutation that is being transformed to I. This permutation depends on the label counts, which in turn depend on the location of the split point. There are $\binom{n+2(m-1)+1}{2(m-1)+1}$ possible ways to assign these, because in addition to one split point we select for both subintervals $(m-1)$ dummy items that mark the empirical counts of labels. Note that here we allow empty intervals. As the above count is the same for both subintervals, we can ignore it.

Hence, the COMPRESS method, given in Table 2, selects the split point that compresses the labels the most. Previously Kononenko [17] has suggested a similar rule for decision trees. Note that COMPRESS has a relation to the entropy:

$$\lg \mathbf{Perm}\,(I) = \lg n! - \sum_{i=1}^{m} \lg n_i!$$

$$H\left(\hat{\boldsymbol{p}}_{\mathrm{ML}}\right) = n \lg n - \sum_{i=1}^{m} n_i \lg n_i.$$

Stirling's formula asserts that $n! \approx \sqrt{2\pi n}(n/e)^n$, and so $\lg n! \approx n \lg n - n \lg e$.

COMPRESS also has a probabilistic justification. Generate $\boldsymbol{p}$ according to a uniform prior distribution for the possible vectors and then generate I from $\boldsymbol{p}$. Then the probability of observing an interval I is

$$\mathbf{P}(I) = \binom{n+m-1}{m-1}^{-1} \binom{n}{n_1, \ldots, n_m}^{-1}, \tag{1}$$

Table 2. COMPRESS: Selection of the best split point

Input: An interval I.
Output: A splitted interval (I_1, I_2) that partitions I.
Algorithm: For each candidate split (I'_1, I'_2) calculate the following value

$$\mathbf{Perm}\left(I'_1\right) \cdot \mathbf{Perm}\left(I'_2\right),$$

and return the candidate with the smallest value.

which follows from the normalizing constant of the Dirichlet distribution [18]. The first part consists of the possible empirical frequencies, and is the same for all frequencies. The second part depends on the empirical frequencies and it is $\lg \mathbf{Perm}\,(I)$. Therefore, if we set the prior probability of a split point to be

$$\mathbf{P}(\text{split at } (I_1, I_2) \mid I) = \frac{\binom{|I_1|+m-1}{m-1}\binom{|I_2|+m-1}{m-1}}{\binom{|I|+2(m-1)+1}{2(m-1)+1}},$$

then we select the same decisions as COMPRESS. Note that $\mathbf{P}(\text{split at } (I_1, I_2) \mid I)$ is a proper probability distribution, because it sums to one.

The numerator in the above equation is $\Theta(|I_1||I_2|)^{m-1}$, in which the dependence on m and $|I|$ is ignored. Hence, COMPRESS maximizes a posterior likelihood that gives more prior probability to splits in the middle of the interval. This is intuitive in the sense that we have more information available for these splits.

4.2 BAY-ENT: Bayesian Estimation of the Real Distribution

Another simple approach is to give an estimate $\hat{\boldsymbol{p}}$ of $\boldsymbol{p}$ and use it to estimate the entropy. We already observed that the maximum likelihood estimate can perform poorly, so let us evaluate a Bayesian estimate of $\boldsymbol{p}$.

First, we note that a good estimate for entropy is not necessarily what we want. For example, one correction, which takes into account part of the bias of the empirical entropy, is the Miller-Madow estimate $\widehat{H}(I) + (m-1)/|I|$ [11]. However, using this estimate gives the same result as using only $\widehat{H}(I)$, because after averaging the entropy estimates we add the same constant to the value of each split.

Let us study a Bayesian estimate for $\boldsymbol{p}$, where we give a prior probability $\mathbf{P}(\boldsymbol{p})$ for each $\boldsymbol{p}$. Then, after observing our data I, we can update our beliefs on the distribution of $\boldsymbol{p}$:

$$\mathbf{P}(\boldsymbol{p} \mid I) \propto \mathbf{P}(I \mid \boldsymbol{p})\,\mathbf{P}(\boldsymbol{p})\,.$$

We choose the uniform prior on $\boldsymbol{p}$, which gives an equal likelihood to *observing* any combination of empirical frequencies, as a corollary to Equation (1).

The posterior is a distribution over $\boldsymbol{p}$, but for simplicity we want a single point estimate. It is known, as Zhu and Lu [19] note, that the Dirichlet prior $\mathbf{P}(\boldsymbol{p}) \propto \prod_{i=1}^{m} p_i^{x_i - 1}$ gives a posterior with the expectation

$$\hat{\boldsymbol{p}}_{\text{BAYES}} = \left(\frac{x_1 + n_1}{x + n}, \ldots, \frac{x_m + n_m}{x + m} \right).$$

Setting the x_is to one we obtain our desired point estimate. Note that we can interpret also this method as maximizing a likelihood, because the entropy $H(\hat{\boldsymbol{p}})$ equals $-\lg \mathbf{P}(\hat{\boldsymbol{p}} \mid \hat{\boldsymbol{p}})$.

If the real $\boldsymbol{p}$ is generated according to the prior that we use, then the estimated posterior gives the true posterior distribution. Hence, in this case $\hat{\boldsymbol{p}}_{\text{BAYES}}$ is a good estimate. Unfortunately, often in practice the prior does not hold. Then, theoretically, relatively little can be guaranteed, except that the posterior converges to the real $\boldsymbol{p}$ with enough samples (if the prior is non-zero everywhere) [20]. In this case Gelman [21] suggests trying several non-informative priors, and trusting the results if they agree.

4.3 CONC: Preferring Non-uniform Intervals Less

Let us also study how changing the objective function from entropy to another concave function affects the performance. Note that any symmetric concave function with a mode at the uniform distribution gives the same ranking for two distributions. However, the decisions differ when pairs of distributions are compared with each other, as is the case with possible splits of an interval. More concave functions prefer a split point with more non-uniform intervals.

One problem that we identified with the empirical entropy was the susceptibility to random noise in the labels. This was caused by the preference to the non-uniform intervals. We suggest the following function, which is nearly linear as a function of distance to the uniform distribution.

$$\text{CONC}(\boldsymbol{p}) = \left(1 - \frac{\|\boldsymbol{u} - \boldsymbol{p}\|_2}{Z} \right)^{1-\epsilon},$$

where $\boldsymbol{u}$ is the uniform distribution, Z is the normalizing value $\max \|\boldsymbol{u} - \boldsymbol{p}\|_2$, and ϵ is a small value, such as 0.99.

As this function resembles 0/1-loss, we expect it to fail in similar conditions. On the other hand, it behaves better near a non-uniform distribution. Hence, it is interesting to see its performance. Note that we use $\text{CONC}(\boldsymbol{p})$ instead of a linear function, because it prefers non-uniform pairs. A linear function would give the same rank for all splits, where the majority label does not change. This is often the case when we empirically estimate the distribution, because the sample size is restricted.

Kearns and Mansour [22] have analyzed in connection of decision trees top-down algorithms that are related to discretization. In fact, these algorithms solve the same problem when restricted to a continuous feature. The authors proved

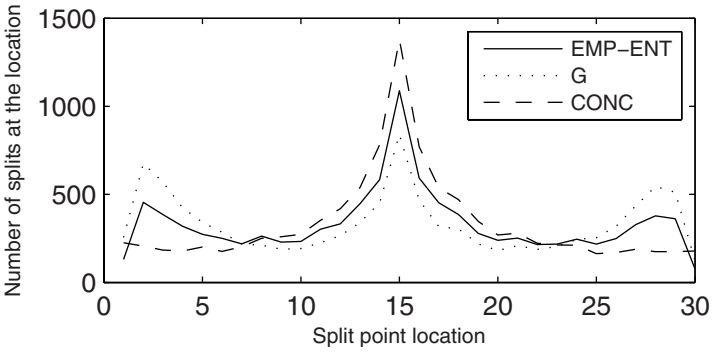

Fig. 5. A quick comparison of functions with different concavity. The distribution that generates the labels and the majority label both change in the middle.

Table 3. Mean squared errors in the tests of Figure 6

	EMP-ENT	COMPRESS	BAY-ENT	CONC
b)	317.50	231.75	272.90	202.91
c)	48.49	44.59	50.60	56.64
d)	15.79	14.25	13.46	11.47
e)	43.89	41.46	41.38	40.51
f)	0.49	0.46	0.44	0.43

formal bounds in the PAC framework. Their analysis suggest using a function that grows faster than the entropy around non-uniform distributions,

$$G(p) = 2\sqrt{p(1-p)}.$$

However, as we have noted, using a concave function is dangerous with small sample sizes. Figure 5 demonstrates this empirically.

5 Empirical Evaluation of the Suggested Solutions

The illustration in Figure 3 gave one example of the behavior of EMP-ENT method, but it does not tell how often we observe, or suffer, from this behavior in practice. We now give results for several different kinds of experiments, and also compare EMP-ENT to the methods put forward in the previous section.

Figure 6 plots the results of six different tests and Table 3 gives the corresponding MSE values. In it MSE is measured from the distance to the correct split point, which is defined by the entropy of the generating distribution. We use MSE in order to penalize more for splitting significantly away from the true location.

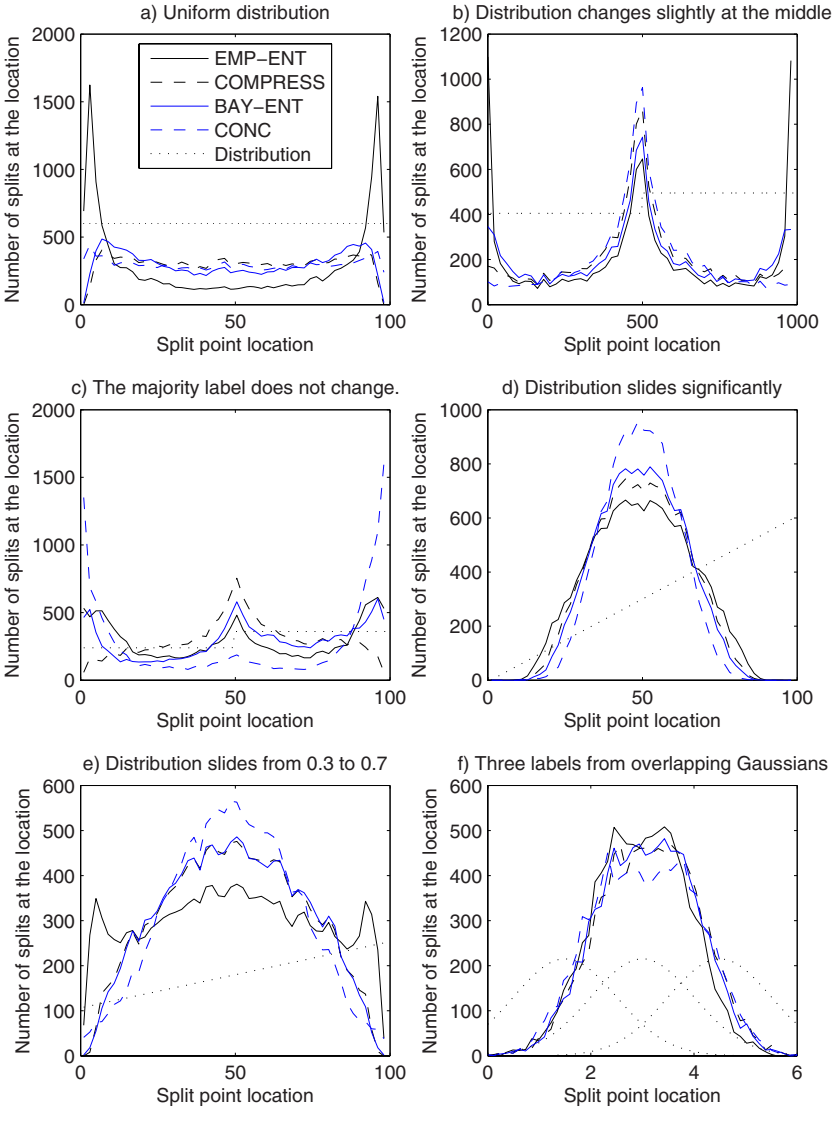

Fig. 6. Comparison of the methods for six different distributions for the interval. In a)-e) the dotted black line is the probability for label a, and in f) the dotted line gives the generating Gaussians for three different labels.

In test a) the intervals are generated from a uniform distribution. Hence, there is no correct place to split, because all subintervals are similar. In the figure we see how EMP-ENT prefers short intervals, and how CONC resists this behavior.

In test b) the generating distribution changes at the middle, first giving probability 0.47 and then 0.53 to the label a. The interval length is 1 000 items.

Although b) has a long interval length, the fact that the distribution changes only slightly, causes EMP-ENT to choose short intervals.

In test c) the distribution also changes at the middle point, but in contrast to test b) the majority label does not. The probability of the label a is first 0.2 and 0.3 after the middle point. Note how CONC fails in this case, although it otherwise performs well. Interestingly, COMPRESS is the only approach to perform well.

In tests d) and e) the generating distribution changes as a linear function of the interval position. In d) the change for the label a is from 0 to 1. In e) the change is from 0.3 to 0.7. The correct split position is in the middle. We see that the more difficult decision in e) causes EMP-ENT to choose also short intervals.

Finally, test f) demonstrates intervals generated from three overlapping Gaussians. Each label — a, b, and c — is generated from a Gaussian distribution with variance one on the unit line. The centers are at 1.5, 3, and 4.5. The correct split position is at 3.

Of course, these results do not guarantee similar performance in a setting that differs significantly from those introduced here. However, we chose these tests to demonstrate the performance in as different circumstances as we could find. They imply that failures with EMP-ENT tend to happen, if the decision is not easy to begin with. Otherwise, the failure rate does not depend on the parameters of the test, such as length of the interval.

6 Implications to Classification: Experimental Evaluation

In this section we study how the suggested solutions affect our motivating application, the NB classifier. We evaluate the splitting methods on 16 commonly used problems from the UCI machine learning repository. We carry out 100 iterations for each problem. During each iteration two-thirds of the data is assigned to a training set and the rest is assigned to a test set. The performances as a probability of predicting the correct label for the test sets are given in Table 4. We also give the number of generated bins for each problem, counted over all features. The BAY-ENT method is omitted, because its performance was close to that of COMPRESS.

We note that the performance difference between EMP-ENT and COMPRESS is negligible. Either the problematic cases that we have discussed do not occur in practice, or due to the nature of either MDL stopping rule or the recursive heuristic the mistakes are not important. In the latter case it could be that MDL is too conservative in its decisions to take advantage in a better splitting, because it maximizes the split probability when using EMP-ENT, as we have noted in [23]. Another potential reason is the use of the recursive heuristic which could hide the problematic decisions. By this we mean that, even if we erroneously split few labels away from an interval, the recursive splitting guarantees that we will also split at the correct place.

Also, CONC performed well, expect in the Bupa domain. Hence, the behavior depicted in test c) of Figure 6 appears not to happen frequently.

Table 4. Performance of splitting methods on Naïve Bayes. The average classification accuracy is over 100 repetitions of randomized training set selection for 16 UCI domains. The average number of bins in each domain is also given.

	Accuracy			Number of bins		
	EMP-ENT	**COMP**	**CONC**	**EMP-ENT**	**COMP**	**CONC**
Iris	94.0	94.0	93.9	6.6	6.6	6.7
Glass	67.8	66.4	67.4	14.4	14.6	13.6
Bupa	62.8	63.5	59.4	6.2	6.3	6.1
Pima	74.0	74.4	74.0	10.3	10.2	11.2
Ecoli	85.1	85.3	85.6	8.2	7.8	7.8
Segmentation	83.2	82.7	82.4	44.4	44.6	44.1
Wine	98.3	98.0	97.3	19.8	19.8	18.9
Australian	85.6	85.5	85.8	14.4	14.4	14.8
German	73.7	73.2	73.8	24.1	24.0	24.9
Iono	88.9	88.5	89.3	88.1	87.5	81.0
Sonar	75.9	76.5	75.9	60.9	59.8	60.5
Wisconsin	97.5	97.5	97.7	17.5	17.6	18.2
Letter	73.7	73.6	73.4	128.8	129.4	129.3
Abalone	58.5	58.5	58.5	41.3	40.9	37.6
Vehicle	59.4	59.1	59.2	44.1	42.6	42.6
Page	93.3	93.2	93.5	46.9	46.1	40.8
Average	79.5	79.4	79.2	36.0	35.8	34.9

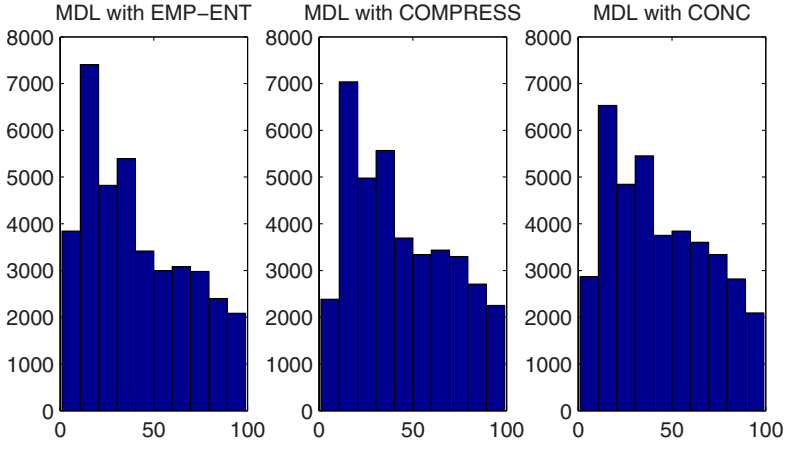

Fig. 7. Distribution of intervals with length less than 100 counted over all domains

We also investigated further what kind of intervals the splitting rules form. The results are given in Figure 7. We see that the EMP-ENT indeed chooses more short intervals than COMPRESS. On average, the number of intervals that COMPRESS selects is slightly lower than for EMP-ENT. This is not surprising, because EMP-ENT maximizes the likelihood of splitting with MDL stopping rule.

7 Conclusions

We gave observations on the behavior of the empirical entropy and noted that for small sample sizes its bias is significant. Then we suggested new methods for choosing the best split point and we empirically evaluated them. A method based on compression fared well in these tests, although the implications were negligible when the splitting methods were applied in the NB classifier. One reason for this behavior could be that the stopping rule and the split point selection method interact in the recursive heuristic. Hence, one possible future direction is to investigate these interactions further.

Acknowledgments

This work has been financially supported by Academy of Finland projects ALEA (210795) and "Machine learning and online data structures" (119699).

References

1. Catlett, J.: On changing continuous attributes into ordered discrete attributes. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 164–178. Springer, Heidelberg (1991)
2. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 1022–1027. Morgan Kaufmann, San Francisco (1993)
3. Raileanu, L.E., Stoffel, K.: Theoretical comparison between the gini index and information gain criteria. Annals of Mathematics and Artificial Intelligence 41, 77–93 (2004)
4. Kohavi, R., Sahami, M.: Error-based and entropy-based discretization of continuous features. In: Simoudis, E., Han, J.W., Fayyad, U. (eds.) Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 114–119. AAAI Press, Menlo Park (1996)
5. Hand, D.J., Yu, K.: Idiot Bayes? not so stupid after all. International Statistical Review 69, 385–398 (2001)
6. Friedman, J.H.: On bias, variance, 0/1-loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery 1(1), 55–77 (1997)
7. Domingos, P., Pazzani, M.J.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–130 (1997)
8. Bouckaert, R.R.: Naive Bayes classifiers that perform well with continuous variables. In: Webb, G.I., Yu, X. (eds.) AI 2004. LNCS (LNAI), vol. 3339, pp. 1089–1094. Springer, Heidelberg (2004)
9. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and unsupervised discretization of continuous features. In: Prieditis, A., Russell, S. (eds.) Proceedings of the Twelfth International Conference on Machine Learning, pp. 194–202. Morgan Kaufmann, San Francisco (1995)
10. Elomaa, T., Rousu, J.: Fast minimum training error discretization. In: Sammut, C., Hoffmann, A.G. (eds.) Machine Learning, Proceedings of the Nineteenth International Conference, pp. 131–138. Morgan Kaufmann, San Francisco (2002)

11. Paninski, L.: Estimation of entropy and mutual information. Neural Computation 15, 1191–1253 (2003)
12. Bialek, W., Nemenman, I. (eds.): Estimation of Entropy and Information of Undersampled Probability Distributions – Theory, Algorithms, and Applications to the Neural Code. Satellite of the Neural Information Processing Systems Conference (NIPS 2003) (2003)
13. Kohavi, R., Wolpert, D.: Bias plus variance decomposition for zero-one loss functions. In: Saitta, L. (ed.) Machine Learning, Proceedings of the Thirteenth International Conference, pp. 275–283. Morgan Kaufmann, San Francisco (1996)
14. Domingos, P.: A unified bias-variance decomposition for zero-one and squared loss. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence, pp. 564–569. MIT Press, Cambridge (2000)
15. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons, New York (1991)
16. Grünwald, P.D.: The Minimum Description Length Principle. MIT Press, Cambridge (2007)
17. Kononenko, I.: On biases in estimating multi-valued attributes. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, pp. 1034–1040. Morgan Kaufmann, San Francisco (1995)
18. Wilks, S.S.: Mathematical Statistics. John Wiley & Sons, New York (1962)
19. Zhu, M., Lu, A.Y.: The counter-intuitive non-informative prior for the Bernoulli family. Journal of Statistics Education 12 (2004)
20. Kass, R.E., Wasserman, L.: The selection of prior distributions by formal rules. Journal of the American Statistical Association 91, 1343–1370 (1996)
21. Gelman, A.: Prior distribution. In: Encyclopedia Environmetrics, vol. 3, pp. 1634–1637. John Wiley & Sons, Chichester (2002)
22. Kearns, M.J., Mansour, Y.: On the boosting ability of top-down decision tree learning algorithms. Journal of Computer and System Sciences 58, 109–128 (1999)
23. Kujala, J., Elomaa, T.: Improved algorithms for univariate discretization of continuous features. In: Kok, J.N., Koronacki, J., López de Mántaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 188–199. Springer, Heidelberg (2007)